

WHAT IS CLAIMED IS:

1. A command processor on a computer system comprising:
  - a graphical user interface for providing a graphical interface to the computer system; and
  - a command interpreter for interpreting commands from a user and for modifying the graphical user interface according to the interpreted commands.
2. The command processor of claim 1 wherein the graphical user interface is modifiable by the user at run time.
3. The command processor of claim 1 wherein the command interpreter interprets user commands to produce graphical objects within the graphical user interface.
4. The command process of claim 3 wherein the command interpreter interprets user commands to assign functionality to the graphical objects.
5. The command processor of claim 1 and further comprising:
  - a suite of integrated circuit design tools, each design tool of the suite having a functionality corresponding to one or

more steps in a design flow process of  
an integrated circuit.

6. The command processor of claim 5 wherein the command processor loads each design tool into the graphical user interface based on user commands.

7. The command processor of claim 1 and further comprising:

a graphics engine tool for drawing contents of a database into the graphical user interface based on a user command.

8. A method of providing a fully customizable graphical user interface comprising:

upon execution of a command processor,  
loading a top level TCL command into a  
namespace;

building graphical objects according to TCL  
commands;

assigning functionality to the built  
graphical objects according to TCL  
commands; and

displaying a user-interactive window  
containing the graphical objects  
according to TCL commands.

9. The method of claim 8 and further comprising:

performing functions based on user interactions with the graphical objects according to their assigned functionality.

10. The method of claim 8 wherein the graphical objects are selected from a group consisting of windows, window panes, buttons, and menus.

11. The method of claim 8 wherein the step of assigning comprises:  
creating a TCL script corresponding to a circuit design function; and  
assigning the TCL script to one of the graphical objects.

12. The method of claim 11 wherein the one of the graphical objects is a button.

13. The method of claim 11 wherein the one of the graphical objects is an item within a pull-down menu.

14. The method of claim 8 and further comprising:  
changing a look and feel of the graphical user interface during a circuit design process.

15. The method of claim 14 wherein the step of changing comprises:

creating new graphical objects using TCL commands; and  
assigning functionality to the new graphical objects.

16. The method of claim 14 wherein the step of changing comprises:

loading a new top level TCL command into the namespace;  
building graphical objects according to the new top level TCL commands;  
assigning functionality to the built graphical objects according to the new TCL commands; and  
displaying the user-interactive window containing the graphical objects according to the new TCL commands.

17. The method of claim 8 wherein before the step of building, the method further comprises:

creating a TCL interpreter object;  
connecting input and output channels; and  
creating room builder objects.

18. The method of claim 8 wherein the steps of building and assigning comprises:

-49-

loading a user specified TCL command configuration script.

19. A method of providing a graphical user interface having no hard coded objects, the method comprising:

loading a top level TCL command into a namespace upon execution of a command processor;  
providing a command interpreter for interpreting commands from a user; and assembling a graphical user interface based on interpreted commands from the user; wherein all objects within the graphical user interface are user defined.

20. The method of claim 19 and further comprising:

changing the graphical user interface based on changing commands from the user; and displaying a changed graphical user interface during operation based on the changing commands.

21. The method of claim 19 and further comprising:

interfacing with a suite of integrated circuit design tools for producing a

-50-

integrated circuit layout and associated netlist.

22. The method of claim 21 wherein the step of interfacing comprises:

loading a design tool from the suite of design tools into the graphical user interface based on a user command.

23. The method of claim 22 wherein the user command is assigned to a graphical object.

24. An integrated circuit software design suite comprising:

a command processor having a graphical user interface and a command interpreter for interpreting user commands, the graphical user interface specified entirely by a user at run time; and one or more design tools corresponding to processes within an integrated circuit design process;

wherein the one or more design tools operate under control of the command processor and within the graphical user interface.